AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS:

1. (Currently Amended) A countermeasure method in an electronic component implementing an elliptical curve type public key encryption algorithm, wherein a point P on the elliptical curve is represented by the projective coordinates (X, Y, Z) such that x=X/Z and y=Y/Z^3, x and y being the coordinates of the point on the elliptical curve in terms of affine coordinates, said curve comprising n elements and being defined on a finite field GF(p), where p is a prime number and the curve has the equation y^2=x^3+a*x+b, or defined on a finite field GF(2^n), with the curve having the equation y^2+x*y=x^3+a*x^2+b, where a and b are integer parameters, the method comprising the steps of:

1) Drawing at random an integer λ such that 0< λ <p;

2) For a point P represented by projective coordinates (X1, Y1, Z1), calculating ~~X'1=l^2*X1, Y'1=l^3*Y1 and Z'1=l*Z1~~ X'1=λ^2*X1, Y'1=λ^3*Y1 and Z'1=λ*Z1, to define the coordinates of the point P'=(X'1,Y'1,Z'1); and

3) Calculating an output point Q=2*P' that is represented by projective coordinates (X2, Y2, Z2).

2. (Previously Presented) A countermeasure method according to Claim 1, wherein the elliptical curve is defined on the finite field GF(p), and the step of calculating Q includes the following steps:

Calculate M=3*X'1^2+a*Z'1^4;

Calculate Z2=2*Y'1*Z'1;

Calculate S=4*X'1*Y'1^2;

Calculate X2=M^2-2*S;

Calculate T=8*Y'1^4; and

Calculate Y2=M*(S-X2)-T.

3. (Currently Amended) A countermeasure method according to Claim 1, wherein the elliptical curve is defined on the finite field GF(p), and further including the following steps:

Drawing at random a non-zero integer ~~1~~ λ of GF(2^n);

Replacing X0 with ~~1~~ λ^2*X0, Y0 with ~~1~~ λ^3*Y0 and Z0 with ~~1~~ λ*Z0;

Drawing at random a non-zero integer ~~1~~ λ of GF(2^n);

Replacing X1 with ~~1~~ λ^2*X1, Y1 with ~~1~~ λ^3*Y1 and Z1 with ~~1~~ λ*Z1; and

Calculating R=P+Q.

4. (Currently Amended) A countermeasure method according to Claim 1, further including the calculation of the projective coordinates of the point R=(X2,Y2,Z2) such that R=P+Q with P=(X0,Y0,Z0) and Q=(X1,Y1,Z1) according to the following steps, with the calculations in each of the steps being effected modulo p:

Replacing X0 with ~~1~~ λ^2*X0, Y0 with ~~1~~ λ^3*Y0 and Z0 with ~~1~~ λ*Z0;

Drawing at random an integer ~~m~~ μ such that 0<~~m~~ μ<p;

Replacing X1 with ~~1~~ λ^2*X1, Y1 with ~~1~~ λ^3*Y1 and Z1 with ~~1~~ λ*Z1;

Calculate U0=X0*Z1^2;

Calculate S0=Y0*Z1^3;

Calculate U1=X1*Z0^2;

Calculate S1=Y1*Z0^3;

Calculate W=U0-U1;

Calculate R=S0-S1;

Calculate T=U0+U1;

Calculate M=S0+S1;

Calculate Z2=ZO*Z1*W;

Calculate X2=R^2-T*W^2;

Calculate V=T*W^2-2*X2; and

Calculate 2*Y2=V*R-M*W^3.

5. (Currently Amended) A countermeasure method according to Claim 1, wherein the elliptical curve is defined on the finite field GF(2^n), where n is a prime number, and the step of drawing a random integer comprises

Drawing at random a non-zero element ~~1~~ λ of GF(2^n).

6. (Currently Amended) A countermeasure method according to Claim ~~1,~~ 5, further including the following steps:

Calculate Z2=X'1*Z'1^2;

Calculate X2=(X'1+c*Z'1^2)^4;

Calculate U=Z2+X'1^2+Y'1*Z'1; and

Calculate Y2=X'1^4*Z2+U*X2.

7. (Currently Amended) A countermeasure method according to Claim 5, further including the following steps, with the calculation in each of the steps being carried out modulo p:

For an input point P=(X0, Y0, Z0), replacing X0 with ~~1~~ λ^2*X0, Y0 with ~~1~~ λ^3*Y0 and Z0 with ~~1~~ λ*Z0;

3) Drawing at random a non-zero element ~~1~~ λ of GF(2^n);

4) For an input point Q = (X1, Y1, Z1), replacing X1 with ~~m~~ μ^2*X1, Y1 with ~~m~~ μ^3*Y1 and Z1 with ~~m~~ μ*Z1; and

5) Calculating R=P+Q.

8. (Currently Amended) A countermeasure method according to Claim 5, further including the following steps:

For an input point P=(X0, Y0, Z0), replacing X0 with ~~1~~ λ^2*X0, Y0 with ~~1~~ λ^3*Y0 and Z0 with ~~1~~ λ*Z0;

Drawing at random a non-zero element ~~m~~ μ of GF(2^n);

For an input point Q = (X1, Y1, Z1) replacing X1 with ~~m~~ μ^2*X1, Y1 with ~~m~~ μ^3*Y1 and Z1 with ~~m~~ μ*Z1;

Calculate U0=X0*Z1^2;

Calculate S0=Y0*Z1^3;

Calculate U1=X1*Z0^2;

Calculate S1=Y1*Z0^3;

Calculate W=U0+U1;

Calculate R=S0+S1;

Calculate L=Z0*W;

Calculate V=R*X1+L*Y1;

Calculate Z2=L*Z1;

Calculate T=R+Z2;

Calculate X2=a*Z2^2+T*R+W^3; and

Calculate Y2=T*X2+V*L^2.

9. (Previously Presented) A countermeasure method according to Claim 1, further including the process of randomizing the representation of a point at the start of the

calculation by the use of a "double and add" algorithm, taking as an input a point P and an integer d, the integer d being denoted d=(d(t),d(t-1),…,d(0)), where (d(t),d(t-1),…,d(0)) is the binary representation of d, with d(t) the most significant bit and d(0) the least significant bit, the algorithm returning as an output the point Q=d.P, according to the following steps:

1) Initialising the point Q with the value P;

2) Replacing Q with 2.Q;

3) If d(t-1)=1 replacing Q with Q+P;

4) For i ranging from t-2 to 0 executing the steps of:

4a) Replacing Q with 2Q;

4b) If d(i)=1, replacing Q with Q+P; and

5) Returning Q.

10. (Previously Presented) A countermeasure method according to Claim 1, further including the process of randomizing the representation of a point at the start of the calculation method and at the end of the calculation method, using a "double and add" algorithm, taking as an input a point P and an integer d, the integer d being denoted d=(d(t),d(t-1),…,d(0)), where (d(t),d(t-1),…,d(0)) is the binary representation of d, with d(t) the most significant bit and

d(0) the least significant bit, the algorithm returning as an output the point Q=d.P, according to the following steps:

1) Initialising the point Q with the value P;

2) Replacing Q with 2.Q;

3) If d(t-1)=1, replacing Q with Q+P;

4) For i ranging from t-2 to 1, executing the steps of:

4a) Replacing Q with 2Q;

4b) If d(i)=1, replacing Q with Q+P;

5) Replacing Q with 2.Q;

6) If d(0)=1, replacing Q with Q+P and;

7) Returning Q.

11. (Previously Presented) A countermeasure method according to Claim 1, further including the following steps:

1) Initialising the point Q with the point P;

2) For i ranging from t-2 to 0, executing the steps of:

2a) Replacing Q with 2Q;

2b) If d(i)=1, replacing Q with Q+P; and

3) Returning Q.

12. (Previously Presented) A countermeasure method according to Claim 1, further including the following steps:

1) Initialising the point Q with the point P.

2) Initialising a counter co to the value T.

3) For i ranging from t-1 to 0, executing the steps of:

3a) Replacing Q with 2Q using a first method if co is different from 0, otherwise using method;

3b) If d(i)=1, replacing Q with Q+P;

3c) If co=0 then reinitialising the counter co to the value T;

3d) Decrementing the counter co; and

4 Returning Q.

13. (Previously Presented) The method of claim 1, wherein said electronic component is a smart card.